Pattern Anal Applic (2011) 14:283–293
DOI 10.1007/s10044-010-0189-3

THEORETICAL ADVANCES

Adaptive kernel approach to the time series prediction

Marcin Michalak

Received: 6 October 2009 / Accepted: 28 October 2010 / Published online: 17 November 2010

Abstract This short article describes two kernel algorithms of the regression function estimation. One of them is called *HASKE* and has its own heuristic of the h parameter evaluation. The second is a hybrid algorithm that connects the SVM and *HASKE* in such a way that the definition of the local neighborhood is based on the definition of the h-neighborhood from *HASKE*. Both of them are used as predictors for time series.

Keywords Time series prediction · Support vector machine · Kernel estimators · Non-parametric regression

1 Introduction

Estimation of the regression function is one of the basic problems that deals with the discipline called machine learning [22, 26]. The aim of the evaluation of regression function is to find some dependencies between variables in the observed dataset. Sometimes these relations can be overt like the dependence between the current intensity and the voltage in the linear element (Ohm's law), but in many cases this dependence is hidden or difficult to notice.

There are two main groups of methods of the regression function estimation: parametrical and non-parametrical methods. Parametrical methods can be described as models with a well-defined functional form with finite number of free parameters, and which values must be established. Usually some optimization criterion is defined and values that optimize it are admitted as "proper". The parametrical model can also be written in the following way:

$$\widetilde{y} = f(x;\theta) \tag{1}$$

where $\widetilde{y}$ is the estimator of the variable y, x is the independent variable and θ is the vector of the model parameters.

Methods from the second group are also described by some free parameters, but the equation of the value estimator should not be concerned as the description of the variable's behavior. Non-parametric estimators do not make any assumptions about the functional form of dependencies between independent and dependent variables. In other words, non-parametric estimator approximates values, but does not try to explain the nature of the dependence. This leads to the general form of the non-parametric estimator:

$$\widetilde{y} = f(x) \tag{2}$$

where the function f is unknown and is the object of the research.

Very common non-parametric regression function estimators are spline functions [4, 9], radial basis functions [13], additive (and generalized additive) models [12], the *LOWESS* algorithm [3] or kernel estimators [17, 28] with support vector machines [1].

The specific kind of data—time series—can be analyzed by the usage of typical methods like autoregressive models, decomposition method or the Fourier analysis. As it will be shown in this article, kernel methods can also be useful as a time series prediction tool, although some standard parts of

M. Michalak (✉)
Institute of Computer Science,
Silesian University of Technology,
ul. Akademicka 16, 44-100 Gliwice, Poland
e-mail: Marcin.Michalak@polsl.pl

M. Michalak
Central Mining Institute, Plac Gwarkow 1,
40-166 Katowice, Poland
e-mail: Marcin.Michalak@gig.eu

algorithms should be changed (the method of the smoothing parameter h evaluation [16].

In this article, author describes two new kernel methods of the time series prediction and both of them are based on the certain regression function estimation. The first method—a kernel estimator *HASKE*—is described based on the Nadaraya–Watson kernel estimator, but can also be based on other estimators from this group. This estimator applies the new adaptive method of smoothing parameter evaluation, using the definition of the h-neighborhood. This new method avoids the big estimation error in a case, where there are no training objects in the neighborhood of the test object. The algorithm is specially designed for time series with the visible periodic dependence between past and present values that is often determined by the nature of the time series. The results of *HASKE* algorithm are compared with other kernel estimators and the well-known decomposition method.

The second kernel predictor—called the *HKSVR*—is a hybrid algorithm that connects the mentioned group of kernel estimators and the support vector machine. It combines the adaptive definition of the test point neighborhood from *HASKE* with advantages of the support vector machine regression, but should not be considered as the extension of *HASKE*. The *HKSVR* is especially designed for the time series for which we cannot point the easy interpretable dependence between past and present values. As the *HKSVR* can be classified as the local support vector regression, its results are compared with that of the support vector machine.

Both algorithms were widely presented in the CORES conference [16].

2 Non-parametric estimators of the regression function

2.1 Kernel estimators

Kernel estimators are the simplest and probably the clearest examples of non-parametric estimators. For example, the Nadaraya–Watson estimator is defined in the following way:

$$\widetilde{f}(x) = \frac{\sum_{i=1}^{n} y_i K\left(\frac{x-x_i}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right)} \tag{3}$$

where $\widetilde{f}(x)$ means the estimator of the $f(x)$ value, n is a number of train pairs (x, y), K is a kernel function and h the smoothing parameter. This estimator assumes that independent (X) and dependent (Y) variables are random variables and the value of the regression function is an approximation of the conditional expected value of the dependent variable Y upon the condition, that the independent variable X took the value x:

$$\widetilde{f}(x) = E(Y|X = x) \tag{4}$$

But more simple interpretation may sound that the estimator of the value $f(x)$ is the weighted average of observed values y_i. Similar kernel estimators are Gasser–Muller [11], Priestley–Chao [27], Stone–Fan [7].

The function must meet some criterions [19] to be used as the kernel function:

1. $\int_{\mathbb{R}} K(u)\,\mathrm{d}u = 1$
2. $\forall x \in \mathbb{R} K(x) = K(-x)$
3. $\int_{\mathbb{R}} uK(u)\,\mathrm{d}u = 0$
4. $\forall x \in \mathbb{R} K(0) \geq K(x)$
5. $\int_{\mathbb{R}} u^2 K(u)\,\mathrm{d}u < \infty$

One of the most popular is the Epanechnikov kernel [6]:

$$K(x) = \frac{3}{4}\left(1 - x^2\right) I(-1 < x < 1) \tag{5}$$

where $I(A)$ means the indicator of the set A. Other popular kernel functions are presented in the Table 1.

The second step in creating the kernel estimator is the selection of the smoothing parameter h. As it is described in [20] and [25], the selection of h is more important than the selection of the kernel function. Small values of h cause the estimator to fit data too much. Big values of the parameter h lead the estimator to oversmooth dependencies in the analyzed set.

The most popular method for the evaluation of the parameter h is the analysis of the approximation of the mean integrated square root error (MISE) that is defined as follows:

$$\mathrm{MISE}(h) = E\left[\int [\widetilde{f}_h(x) - f(x)]^2\,\mathrm{d}x\right] \tag{6}$$

The MISE can be expressed also as the sum of integrated variance (IV) and integrated squared bias (ISB):

$$\mathrm{MISE}(h) = \int \mathrm{Var}\,\widetilde{f}_h(x)\,\mathrm{d}x + \int \mathrm{Bias}^2\,\widetilde{f}_h(x)\,\mathrm{d}x \tag{7}$$

The approximation of ISB and IV can be expressed as:

$$\mathrm{ISB} = \frac{1}{4}\sigma_K^4 h^4 R(f'') + O(h^6) \tag{8}$$

$$\mathrm{IV} = \frac{R(K)}{nh} - \frac{R(f)}{n} + O\left(\frac{h}{n}\right) \tag{9}$$

Table 1 Popular kernel functions

Uniform	$K(x) = \frac{1}{2} I(-1 < x < 1)$
Triangular	$K(x) = (1 - \lvert x\rvert)\, I(-1 < x < 1)$
Biweight	$K(x) = \frac{15}{16}(1 - u^2)\, I(-1 < x < 1)$
Gaussian	$K(x) = \frac{1}{\sqrt{2\pi}} \exp -u^2/2$

where $R(L) = \int_{-\infty}^{\infty} L^2(x)\,\mathrm{d}x$ and $\sigma_L^4 = \int_{-\infty}^{\infty} x^k L(x)\,\mathrm{d}x$. That leads to the approximation of MISE (AMISE) given by the equation:

$$\mathrm{AMISE}(h_0) = \frac{5}{4}[\sigma_K R(K)]^{\frac{4}{5}} R(f'')^{\frac{1}{5}} n^{-\frac{4}{5}} \tag{10}$$

Optimization of the MISE in respect of h gives:

$$h_0 = R(K)^{1/5}\left(\sigma_K^4 R(f'')\right)^{-1/5} n^{-\frac{1}{5}} \tag{11}$$

The value of the expression σ_K^4 depends on the kernel function K, but the value of $R(f(x))$ is unknown, so it is often replaced by some estimators. It leads to the following expression:

$$h_0 = 1.06 \min(\widetilde{\sigma}, \widetilde{R}/1.34) n^{-\frac{1}{5}} \tag{12}$$

Details of derivations can be found in [20]. More advanced methods of h evaluation can be found in [7, 10, 23–25].

2.2 Support vector machines

Support vector machines (SVM) were defined in [1] and later in [18, 26].

Although it was invented as a classification tool, SVM is also used for the regression problem [5]. In the linear version of this method, the estimated function $f(x)$ is the linear combination of the vector of independent variables:

$$\widetilde{f}(x) = w'x + w_0 \tag{13}$$

defining also a margin ε as a precision of the estimation. To avoid the overfitting of the regression function some slack variables ξ_i, ξ_i^* are also introduced. All these assumptions lead to the definition of the vector w, that should minimize the following criterion (n is the number of training objects):

$$J(w, \xi) = \frac{||w||^2}{2} + C\sum_{i=1}^{n}\left(\xi_i + \xi_i^*\right) \tag{14}$$

with constraints:

$$\begin{cases} y_i - wx_i - w_0 & \leq \varepsilon + \xi_i \\ wx_i + w_0 - y_i & \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* & \geq 0 \end{cases} \tag{15}$$

The constant $C > 0$ determines the trade-off between the flatness of $\widetilde{f}$ and the amount up to which deviations larger than ε are tolerated [21].

For each train object the pair of the Lagrange multipliers α_i, α_i^* are obtained. Then, the value of the regression function can be calculated as:

$$\widetilde{f}(x) = \sum_{i=1}^{n} wx + w_0 \tag{16}$$

where

$$w = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)x_i \tag{17}$$

$$w_0 = -w'(x_r + x_s)/2 \tag{18}$$

and where x_r and x_s are support vectors (the notion explained in the next paragraph).

Now it is noticeable that not all of the training vectors take part in the evaluation of the regression value. Only vectors x_i which Lagrange parameters $\alpha_i - \alpha_i^* > 0$ influence the result and these vectors are called support vectors.

There also exists the non-linear version of SVM and it differs in such a way that the scalar product of two vectors is replaced by the function that performs the corresponding product in higher-dimensional space. Detailed calculations can be found in [21].

This model of the support vector regression (SVR) is the global one, but there are also a number of its local modifications [8, 14].

3 The modification of the data space

Typical time series data can be described as the set of pairs (t, x_t), where t is the time variable and x is the observed variable. For the purpose of kernel time series prediction a small transformation of the data space must be performed. Let us assume that there is a parameter $p_m \in \mathbb{N}$ defining the maximal prediction horizon that is our interest. Then, the original set of pairs (t, x_t) is transformed to the set of pairs (x_t, x_{t+p_m}). This transformation decreases the number of pairs from n in the original dataset to $n - p_m$ in the transformed data.

Figure 1 shows the same time series in original data space and in the modified one. This series is taken from [2] and is marked as G.

The G series describes the increase in the number of American airlines passengers per month (in thousands) between 1949 and 1960. From its nature it is intuitive to set p_m equal to 12.

The transformation of the time series modifies the task of the time series prediction and leads to the estimation of the regression function in the modified space. The prediction of the value of the time series in the moment t (x_t) is equivalent to the evaluation of the value of the regression function for the argument $x_{t-p_m}(\widetilde{f}(x_{t-p_m}))$.

It is typical that time series are not divided into train and test subsets. The predictor is trained on the basis of the historical data and verified on the basis of the present data. If the maximal interesting prediction horizon is p_m and the historical data are from x_1 to x_k it is suitable to verify the

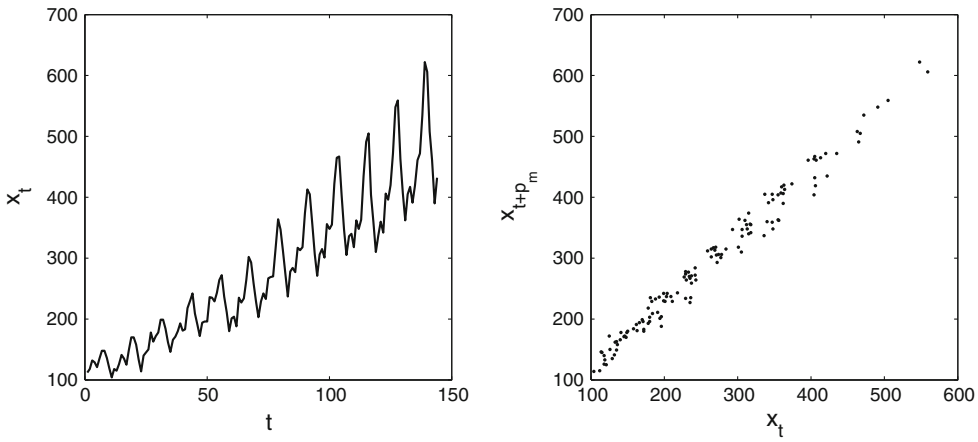

Fig. 1 The same series in two data spaces

prediction model in the following way: observations x_1 to x_{k-p_m} become the train set and the rest becomes the test set.

In the modified space, the division into train and test set comes in the analogical way. The whole dataset is the set of pairs $\{(x_1, x_{1+p_m}), (x_2, x_{2+p_m}), \ldots, (x_{n-p_m}, x_n)\}$. Then, last p_m pairs should become the test set and rest the train set:

$$\begin{aligned}\text{train} &= \{(x_1, x_{1+p_m}), (x_2, x_{2+p_m}), \ldots, (x_{n-2\cdot p_m}, x_{n-p_m})\}\\ \text{test} &= \{(x_{n-2\cdot p_m+1}, x_{n-p_m+1}), (x_{n-2\cdot p_m+2}, x_{n-p_m+2}), \ldots,\\ &\quad (x_{n-p_m}, x_n)\}\end{aligned}$$

Some adaptive algorithms need also a special subset of train data called tune set. The algorithm procedures evaluate adaptive parameters on the reduced train set and verify them on the tune set. The division of the train set into the smaller train set and the tune set is performed as the division of the dataset into the train and tune set—last p_m pairs of the train set become the tune set. The result of the double division can be simply defined as follows:

$$\begin{aligned}\text{train} &= \{(x_1, x_{1+p_m}), (x_2, x_{2+p_m}), \ldots, (x_{n-3\cdot p_m}, x_{n-2\cdot p_m})\}\\ \text{tune} &= \{(x_{n-3\cdot p_m+1}, x_{n-2\cdot p_m+1}), (x_{n-3\cdot p_m+2}, x_{n-2\cdot p_m+2}), \ldots,\\ &\quad (x_{n-2\cdot p_m}, x_{n-p_m})\}\\ \text{test} &= \{(x_{n-2\cdot p_m+1}, x_{n-p_m+1}), (x_{n-2\cdot p_m+2}, x_{n-p_m+2}), \ldots,\\ &\quad (x_{n-p_m}, x_n)\}\end{aligned}$$

4 Problems with the typical kernel prediction

As it was mentioned in Sect. 2.1 kernel estimators need two significant elements: kernel function and the smoothing parameter. The choice of the kernel function is not as significant as the choice of the smoothing parameter. The simplest formula of the optimal h value is the Eq. 12. we need to evaluate it on the basis of the train set. In this case we do not need the tune set, so we can treat the sum of the train and tune set as the train set.

The simple experiment of prediction of 12 values of the *G* series, with the usage of Nadaraya–Watson kernel estimator (3), shows the basic problem of the kernel time series prediction. In Table 2 some estimated values are zero what causes very big prediction error (values typed with bold font).

It occurs when the denominator of Eq. 3 equals zero and it makes the division unrealizable. It means also that there is no pair of train and test values that $K(x_{\text{train}}, x_{\text{test}}) > 0$. This situation is easier to describe if we define the notion of h-neighborhood.

The set h_{x_i} is the h-neighborhood of the test point x_i if and only if it satisfies the condition:

$$h_{x_i} = \left\{x_t \in \text{train} \;:\; K\left(\frac{x_t - x_i}{h}\right) > 0\right\} \tag{19}$$

From this point of view we can say that the big prediction error is caused by the empty h-neighborhood of some test points. It may seem correct to increase the support of the kernel function by the increase of the smoothing parameter value. However, from the other hand we know that increasing the value may cause the other unwanted effect—the oversmoothing. The algorithm that gives us a compromise between the non-empty h-neighborhood and the oversmoothing is the *HASKE* algorithm, described in Sect. 5.

Table 2 The result of 12 values predicted by Nadaraya–Watson

t	x_t	$\widetilde{f}_{NW}(x_{t-12})$	absolute error
I 60	417	389	28
II 60	391	377	14
III 60	419	448	29
IV 60	461	442	19
V 60	472	451	21
VI 60	535	514	21
VII 60	622	0	**622**
VIII 60	606	0	**606**
IX 60	508	501	7
X 60	461	449	12
XI 60	390	390	0
XII 60	432	447	15

5 *HASKE* algorithm

5.1 Background

Performing time series prediction as the kernel estimation of the regression function may meet the problem of empty h-neighborhood for test objects. It occurs that typical algorithms of smoothing parameter evaluation fail in respect to time series prediction. Results of experiments—some of them are shown in Table 2—suggest to modify the value of smoothing parameter in such a way that for every test object its h-neighborhood would be non-empty.

Heuristic Adaptive Smoothing Parameter Kernel Estimator Algorithm (*HASKE*) solves the mentioned problem. The solution is the set of two parameters μ and α. Each of them depend on the given time series, but only the first of them is connected with the problem of the empty h-neighborhood. For the given time series its training part is divided into two separate subsets: train and tune. Then the Nadaraya–Watson kernel estimator is trained with respect to the value of the μ, and the error of the tune set prediction is observed. The value that gives the lowest prediction error on the tune set is chosen as the optimal value of the μ parameter.

HASKE, like other mentioned kernel estimators, is not defined for typical time series space (pairs of observation and time stamp: (t, x_t)), but requires the transformation to the new space with clearly defined dependent and independent variables. The typical form of transformation is definition of the time interval Δt, that implies the new set of pairs of observation $(x_t, x_{t+\Delta t})$. This assumption requires setting the Δt value, that is usually prediction horizon or the strongest time series period length. *HASKE* should not be applied when Δt is hard to determine or when the correlation between dependent and independent variables is low.

5.2 Definition

Let us define the μ parameter that modifies kernel regression in the following way:

$$\widetilde{f}(x) = \frac{\sum_{i=1}^{n} y_i K\left(\frac{x-x_i}{\mu h}\right)}{\sum_{i=1}^{n} K\left(\frac{x-x_i}{\mu h}\right)} \tag{20}$$

It means that the original value of the smoothing parameter is multiplied by the μ value. The new value of smoothing parameter will be written as $h_\mu = \mu h$. Now we can start to increase the μ value and observe the error on the test set. The result of this simple experiment on the G series is shown on Fig. 2. First two steps down are connected with the fact that the h-neighborhood for some test point becomes non-empty. Increasing the μ value further causes the effect of oversmoothing.

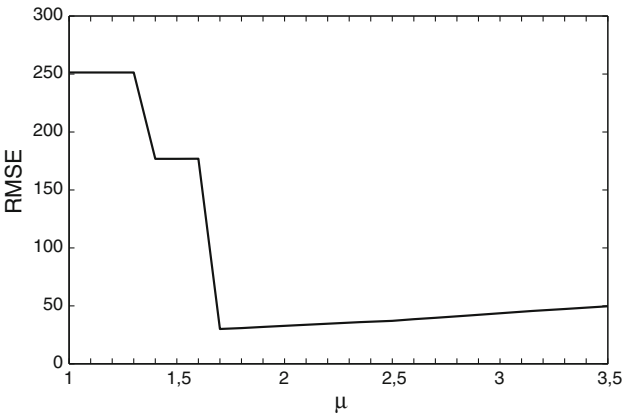

Fig. 2 Dependence the RMSE on μ

It is very important to set the question: How far shall we increase the μ parameter? If we do it in an arbitrary way it may occur that this method gives more damage than profits. This value should be data dependent, so we should evaluate it in the adaptive way. That is why the train set was divided into the smaller train and the tune set. We can observe from Fig. 2 how the tune set prediction error changes by the action of μ changes.

To assure the independence of the μ value from the phase of the time series period, it is evaluated as the median of μ_i values, evaluated for every phase of the series period. The phase as itself is considered by the author as the following notion: let us consider the time series determined by the values from t_0 to t_k and the prediction horizon p_m. For the assumed prediction horizon p_m it is possible to define $M = p_m$ phases, with indices $ph = 0, 1, \ldots, M-1$, as experiments that are defined as the prediction of p_m values on the basis of values from t_0 to t_{k-ph}.

Let us assume that rmse(t, p, h) is the prediction error of p consecutive time series values, from the time interval $[t+1, t+p]$, with the usage of the h smoothing parameter value. If p_m means the maximal interesting prediction horizon then the formula of the adaptive μ value will become:

$$\mu = \text{med}\{\arg\min_{\mu} \text{rmse}(t-i, p_m, h\cdot\mu), \quad i = 0, 1, \ldots, p_{\max} - 1\} \tag{21}$$

The final μ value is 2.4 and the three-dimensional chart is shown in Fig. 3 which shows the dependence of the prediction error on the phase of the period and the μ value. It is also worth to see the quality of the solution against the background of the error on the test set. Figure 4 shows the same dependence as that of Fig. 2. The solid line represents how the prediction error depends on the value of the μ parameter. There is a local minimum at $\mu = 1.5$. The point indicates the result of adaptive μ parameter evaluation and

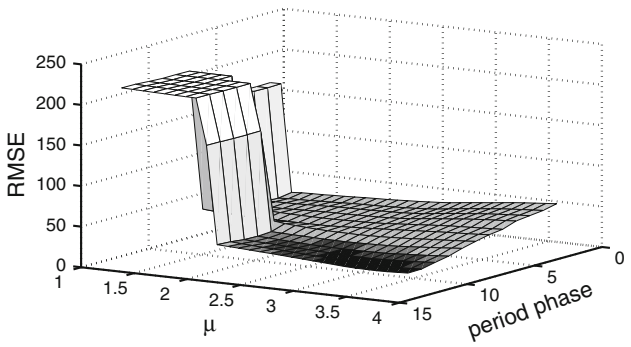

Fig. 3 Dependence the RMSE on μ and the period phase

it is easy to notice that the error obtained with the adaptively obtained value is not significantly higher than the global minimal prediction error.

As we see in Fig. 4, the heuristic of μ parameter evaluation avoids the effect of empty h-neighborhoods. On the other hand, this heuristic also cause a small effect of oversmoothing. That is why another adaptive parameter should be defined—it is called underestimation. The underestimation α of the real value y, when its estimator is $\widetilde{y}$ is defined as follows:

$$\alpha = \frac{\widetilde{y}}{y}$$

In this case we can use the tune set again and evaluate the underestimation after the adaptive value of μ was obtained. The final underestimation is assumed as the median of all underestimations on the tune set:

$$\alpha = \operatorname{med} \alpha_i, \quad i = 1, \ldots, p_m$$

If the error is symmetric (its mean value is nearly zero) then α is almost 1 and does not improve the prediction result, but if the error is not symmetric, the usage of this parameter gives better results. From the other hand, if the error is not symmetric, we can observe the underestimation (overestimation) of the predicted value, as the alpha

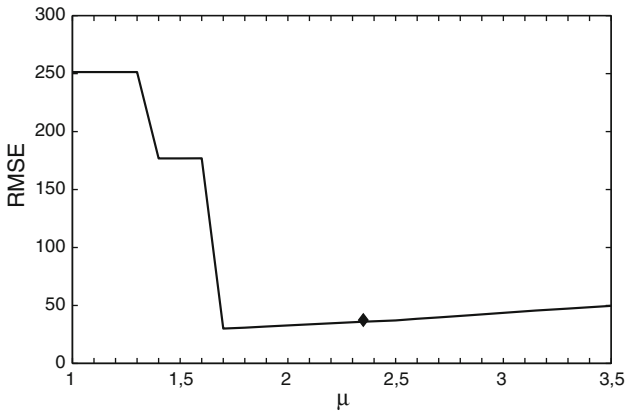

Fig. 4 Dependence the RMSE on μ with the heuristic obtained value

becomes different from one. If α becomes bigger than 1 then it means that predicted values are higher than real ones. Therefore, it suggests to decrease the predicted values with the fraction equals α. If α becomes lower than 1 it means that predicted values are lower than real ones, so it suggests to increase predicted values with the same fraction.

Table 3 shows the influence of the parameters μ and α on the prediction error. It also shows the results of prediction of the last period of G series with the usage of Nadaraya–Watson estimator (NW), *HASKE* only with the μ parameter evaluated ($HASKE_{\mu}$) and full *HASKE* ($HASKE_{\mu,\alpha}$).

Including μ and underestimation α to the final prediction formula we obtain:

$$x_{t+p_m} = \widetilde{f}(x_t) = \frac{1}{\alpha} \frac{\sum_{i=1}^{m-p_m} y_i K\left(\frac{x_i - x_t}{h \cdot \mu}\right)}{\sum_{i=1}^{m-p_m} K\left(\frac{x_i - x_t}{h \cdot \mu}\right)} \tag{22}$$

All the steps of *HASKE* algorithm can be described as follows:

1. Define the maximal interesting prediction horizon p_m.
2. Transform time series from the (t, x_t) space to the (x_k, x_{k+p_m}) space.
3. Split the obtained set of pairs into the train and tune set. Last p_m pairs of the initial set become the tune set, the rest remain in the train set.
4. Define the maximum value of μ ($\mu_{\max}$) and the step of the μ increase ($\Delta\mu$). Then, for each phase of the prediction horizon $\mathrm{ph} = 0, 1, 2, \ldots, p_m - 1$ do the following:
   - For $i = 1$ to $i = \frac{\mu_{max}-1}{\Delta\mu}$ observe the prediction error on the phase tune set $\mathrm{rmse}_{\mathrm{ph}}$.

Table 3 The improvement of prediction obtained with the usage of μ and α in *HASKE*

t	x_t	NW(x_{t-12})	HASKE$_{\mu}$(x_{t-12})	HASKE$_{\mu,\alpha}$(x_{t-12})
I 60	417	389	382	411
II 60	391	377	367	395
III 60	419	448	421	453
IV 60	461	442	412	444
V 60	472	451	437	471
VI 60	535	514	493	531
VII 60	622	0	553	595
VIII 60	606	0	555	598
IX 60	508	501	487	524
X 60	461	448	421	454
XI 60	390	390	383	412
XII 60	432	447	420	452
RMSE		275.26	37.39	17.18

- Select the minimal value of the phase prediction error rmse_{ph}. The argument μ_{ph} is the argument of the minimal rmse_{ph} value.

5. The median of $\mu_{\text{ph}}, \text{ph} = 0, 1, \ldots, p_m - 1$ values becomes the μ value.
6. Find underestimations of all tune objects, as the result of the prediction with the usage of μ value and take median of them as the α value.
7. Perform the *HASKE* prediction as in Eq. 22.

6 The *HKSVR* estimator

6.1 Background

The support vector regression model described in Sect. 2 is a global one. There is also a number of its modifications that use a local learning paradigm. The algorithm presented in [8] uses *kNN* as a local training set. The other algorithm says that the value of the ϵ parameter depends on local covariance matrix (Σ_i), calculated on the basis of training points from the neighborhood of point x_i [14].

The heuristic of the smoothing parameter evaluation, presented with the *HASKE* algorithm, gives more appropriate definition of the test point neighborhood because it makes it possible to reduce the time series prediction error. It is worth to check, whether that new definition of neighborhood improves the results of the support vector regression for time series.

The Hybrid, Kernel and Support Vector Regression algorithm (*HKSVR*) combines the kernel regression (considered as Nadaraya–Watson estimator or similar) and SVM regression. Initial step of the algorithm determines the neighborhood of train objects for each test point. Second step performs support vector regression for the test point on the basis of its train neighbors.

Similarly, as *HASKE*, the *HKSVR* performs prediction as the estimation of the regression function in the modified space. That means that the value of the parameter that defines the transformation is necessary. As distinguished from *HASKE*, the *HKSVR* is designed for prediction of time series, where it is hard to point their period length. The length of the time series period may be determined with the usage of the Fourier transformation.

It is important to notice that the *HKSVR* usefulness depends on the correlation of data in modified space. Usage of the *HKSVR* brings the prediction improvement if the correlation is significant (close to one).

6.2 Definition

As it was mentioned in the previous section, this paper describes the new local estimator. It is based on the h-neighborhood definition and its adaptive evaluation, but is not the extension of *HASKE*. First step of the algorithm is the choice of the parameter δ that defines the transformation of the time series from its original space to the space defined in Sect. 3 As it was mentioned in Sect. 1, the *HKSVR* is dedicated for time series with "hidden" dependence between its previous values. The parameter δ, that determines the transformation to the modified space, does not have to be connected with the maximal interesting prediction horizon and that is why the other denotation is used. The δ value can be evaluated with the usage of the Fourier analysis and δ is the length of one of the harmonics.

In the second step, all data are divided into train, tune and test set. Then the adaptive value of the μ parameter is evaluated, as it is performed in the *HASKE* algorithm. After that, for every test object its h_μ-neighborhood is determined and becomes the train set for the support vector regression. Finally, the prediction is performed as the local SVR.

More detailed steps of the *HKSVR* algorithm are as follows:

1. Split train data into the tune set and the smaller train set.
2. Evaluate smoothing parameter h in the standard way (for example Eq. 12).
3. Find values of the μ and α as it takes place in the *HASKE* algorithm for the task of prediction of the tune set.
4. For every test object:
   (a) find its h_μ-neighborhood (Eq. 19) in the whole train set—it becomes the train set for the *SVR*,
   (b) learn the support vector machine,
   (c) find the value for the test object as the result of local support vector regression,
   (d) divide the result by the underestimation (α).

7 Time series prediction

All algorithms were performed on synthetic and real data. Besides, all of experiments corresponded to the rule of unbiased prediction. This rule assumes that the prediction value is equal to the expected value of the predicted random variable. It is required that expected value of the difference between the predicted random variable and the prediction value leads to zero with the increase of the amount of the training data [30].

As the measure of ex post prediction error the Root Mean Squared Error (RMSE) was used:

$$\text{err} = \frac{1}{k}\sqrt{\sum_{i=1}^{k}(y_i - \widetilde{y}_i)^2} \tag{23}$$

where k is the number of objects in the test set or the maximal interesting prediction horizon.

7.1 *HASKE* results

Four time series were used for experiments with the *HASKE* algorithm. Two of them, marked as *G* and *E* are the real ones, taken from [2]. The first one describes monthly number of passengers, flying international airlines in USA between January 1949 and December 1960. The second one represents the yearly number of sunspots between 1700 and 1991. The remaining two series are synthetic ones, defined in the following way:

$$M(x) = 7.274\mathrm{e}^{0,0184x} + 9\sin\left(\frac{x}{3}\right) - 4\sin\left(\frac{x}{10}\right) + 11 + \epsilon, \quad x = 1, 2, \ldots, 150 \tag{24}$$

$$N(x) = T(x) + 8\sin\left(\frac{2\pi x}{10}\right) + \epsilon, \quad x = 1, 2, \ldots, 80 \tag{25}$$

where $T(x)$ is a trend:

$$T(x) = \begin{cases} \frac{1}{2}x & x \leq 40 \\ 2x & x > 40 \end{cases} \tag{26}$$

and ϵ is the uniform noise on the interval $[-0.5, 0.5]$.

The *HASKE* algorithm was compared with three kernel estimators: Stone–Fano (SF), Gasser–Muller (GM), Nadaraya–Watson (NW), and the decomposition method [15]. Models with linear and exponential trends were considered and additive and multiplicative models were also checked. The comparison of eight models (four of the decomposition method) is shown in the Table 4 (best results are type with bold font). For every series the prediction of the one full period was performed. The prediction error (RMSE) was calculated.

7.2 The *HKSVR* results

The hybrid model, connecting *HASKE* and the *SVR*, was used as a financial time series prediction tool for the Warsaw stock index *WIG*20 closing values (WIG 20, 2007). Sample of this series was taken from 10 April 2003 to 23 July 2007 and is shown in Fig. 5.

Additionally, the rate of return time series was calculated. The rate of return of the time series $x(t)$ was defined as:

$$r_x(t) = \frac{x_t - x_{t-1}}{x_{t-1}} \tag{27}$$

and the corresponding time series was marked as *rWIG*20.

The prediction horizon p varied from 1 to 10 days. The δ parameter was evaluated as the length of the nth maximal harmonic of the Fourier transformation. The length of the first harmonic for time series was comparable to the length of the time series. As a result of this fact, the first harmonic length was not considered as the value of the δ parameter.

The results of the experiment for the *WIG*20 series, showing the dependence of the estimation error improvement, rounded to the integer value, are shown in Table 5. The improvement is defined as the difference between the SVR error and the *HKSVR* error.

These results are also statistically described. Table 6 shows the average prediction improvement and its standard deviation. The averaging took place through nine considered Fourier harmonics as the basis of the δ-based time series transformation.

The character of the prediction improvement can be more visible with the definition of the ρ coefficient, that is the quotient of the average prediction improvement and its standard deviation. Observing the value of that coefficient may help to decide whether the usage of the *HKSVR*

Fig. 5 *WIG20* time series

Table 4 Comparison of the decomposition method, kernel estimators and *HASKE*

Decomposition					SF	GM	NW	*HASKE*
Trend	Exp.		Lin.					
Model	Add.	Mult.	Add.	Mult.				
M series	13.36	23.23	28.47	33.68	31.24	58.53	42.77	**8.74**
N series	21.99	50.69	33.65	49.22	43.81	75.30	49.09	**4.6**
G series	40.32	26.60	64.63	68.52	139.01	475.64	275.26	**17.18**
E series	72.06	77.72	72.56	77.57	33.66	301.30	**33.37**	36.76

Table 5 The dependence of the *WIG*20 estimation accuracy increase on the prediction horizon and the harmonic

*n*th Maximal harmonic	Prediction horizon *p*									
	1	2	3	4	5	6	7	8	9	10
2	−91	−40	−124	−117	−53	−149	−25	25	3	1
3	−1	23	6	14	4	14	91	45	19	16
4	−298	13	40	25	11	69	48	19	−8	−41
5	−46	112	−19	−17	−11	−10	−24	−27	−20	−47
6	−44	−24	3	17	26	21	40	0	−146	−11
7	−13	4	−34	−28	6	−2	−5	−4	0	0
8	−2	3	0	−11	266	208	189	136	104	102
9	−33	−49	403	347	−6	0	0	0	0	0
10	114	152	−143	0	0	−85	0	0	0	0

Table 6 Statistical description of the *WIG*20 estimation improvement with the usage of the *HKSVR* model

Horizon	1	2	3	4	5	6	7	8	9	10
Avg	−45.9	21.7	14.6	25.6	26.9	7.4	35.0	21.7	−5.4	2.3
SD	109.6	67.6	158.0	127.7	92.4	98.7	69.2	47.7	63.9	42.9

improves the prediction or not. The higher the positive values the higher is the improvement.

Table 7 shows the ρ values of the *WIG*20 and *rWIG*20 time series prediction improvement compared. All positive values of the ρ are bold and mean that the usage of the *HKSVR* gave the prediction improvement.

It can be noticed that the improvement of the rate of return time series prediction decreased significantly. Majority of positive ρ values became negative.

7.3 ρ coefficient normalization and results interpretation

The definition of the ρ parameter causes that it is not normalized to the [0,1] interval and original ρ values are from the $[-\infty, \infty]$ interval. The assumption of the normalization formula $Q(\rho)$ is as follows:

- $Q(\rho) = 0$ is the asymptotic worst value and corresponds to $\rho = -\infty$
- $Q(\rho) = 1$ is the asymptotic best value and corresponds to $\rho = \infty$
- $Q(\rho) \in (-\infty, 0.5)$ corresponds to prediction worsening
- $Q(\rho) \in (0.5, \infty)$ corresponds to prediction improvement
- $Q(\rho) = 0.5$ means that there is no improvement and it corresponds to $\rho = 0$
- $1 - Q(-\rho) > Q(\rho)$, for $\rho > 0$, that for small $|\rho|$ values the worsening has stronger influence on the Q value that the improvement (for example: $Q(0.01) = 0.51$ and $Q(-0.01) = 0.3$).

Most of these assumptions are satisfied by the sigmoid unipolar function:

$$Q(\rho) = \frac{1}{1 + e^{-\beta\rho}} \tag{28}$$

with $\beta \in (0, 1]$. Figure 6 shows the $Q(\rho)$ for three values of β: $\beta = 0.1$ (solid line) $\beta = 0.2$ (dashed line) and $\beta = 0.4$ (dot–dashed line).

We may see that the constant value of β causes that the last condition for Q function is not fulfilled. But we also see that the value $\beta = 0.4$ seems to be proper for $\rho \in (-\infty, 0)$ and the value $\beta = 0.1$ seems to be proper for $\rho \in (0, \infty)$. If we define values β_- for $\rho \in (-\infty, 0)$ and β_+ for $\rho \in (0, \infty)$, the $\beta(\rho)$ can be also the sigmoid unipolar function, with a small modification:

$$\beta(\rho) = -\frac{\beta_- - \beta_+}{1 + e^{-\rho}} + \beta_- \tag{29}$$

Table 7 The ρ coefficient values for *WIG*20 and *rWIG*20 series

p	1	2	3	4	5	6	7	8	9	10
ρ_{WIG20}	−2.4	**3.1**	**10.8**	**5.0**	**3.4**	**13.4**	**2.0**	**2.2**	−11.9	**18.9**
ρ_{rWIG20}	−4.9	−1.6	−1.3	−1.8	−2.6	**3.6**	**23.9**	−2.1	**4.9**	−4.3

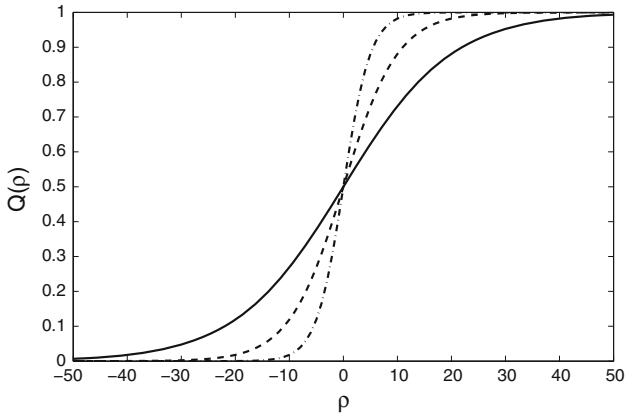

Fig. 6 $Q(\rho)$ function for several values of β

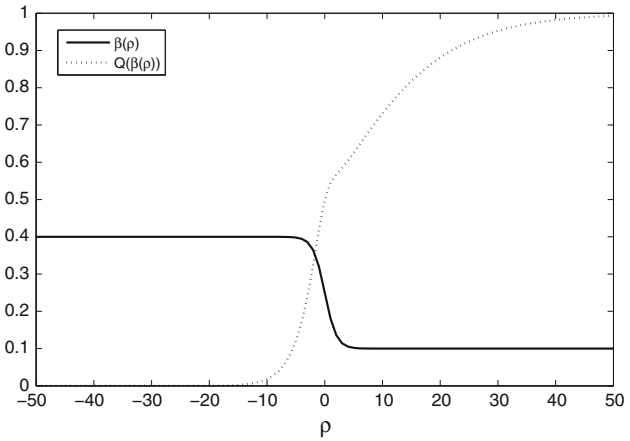

Fig. 7 $Q(\rho)$ and $\beta(\rho)$ functions for $\beta_- = 0.4$ and $\beta_+ = 0.1$

The β_- parameter describes the decrease of the Q function for ρ values smaller than 0 and the β_+ parameter describes the increase of the Q function for ρ values greater than 0.

Applying Eq. (29) to Eq. (28) we finally obtain:

$$Q(\rho) = \frac{1}{1 + e^{-\rho\left(\frac{\beta_+ - \beta_-}{1+e^{-\rho}} + \beta_-\right)}} \tag{30}$$

Figure 7 shows the function $Q(\rho)$ (black dotted line) and $\beta(\rho)$ (black solid line) for specified values of "betas".

Table 8 shows normalized Q values of prediction improvement for 10 prediction horizons and for both time series. Values that mean the positive effect of using the *HKSVR* instead of the *SVR* are marked in bold font.

These results may be surprising. But it is worth to remind that the *HKSVR* model bases on the *HASKE* estimator and this estimator is trying to find the regression function in the modified space (strongly connected with the analyzed time series). In this case, when the points in the modified space exhibit a correlation (more specifically: there is a correlation between the dependent and independent variable), the kernel estimator is able to approximate the regression function.

So let us examine correlations between the dependent and independent variables in the *HKSVR* model for different prediction horizons and for the following harmonics (from 2nd to 10th). Tables 9 and 10 show how the correlation of dependent and independent variables changes depending on the prediction horizon and the harmonic (h_i: the ith highest Fourier harmonic).

We see that the *WIG*20 series has significant correlation in the modified space and high values of correlation do not depend on the used order of the harmonic that defines the translation value δ. Analogous correlations for the *rWIG*20 series are insignificant (very close to zero). So it should be claimed that the usage of the *HASKE* estimator and the *HKSVR* model is justified in cases where the correlation between the dependent and independent variables is significant.

Table 8 The Q coefficient values for *WIG*20 and *rWIG*20 series

p	1	2	3	4	5	6	7	8	9	10
Q (*WIG*20)	0.29	**0.59**	**0.75**	**0.62**	**0.59**	**0.79**	**0.57**	**0.57**	0.01	**0.87**
Q (*rWIG*20)	0.12	0.36	0.39	0.34	0.27	**0.60**	**0.92**	0.32	**0.62**	0.15

Table 9 Correlations for the *WIG*20 time series in the modified space

p	1	2	3	4	5	6	7	8	9	10
h_2	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.91	0.91
h_3	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
h_4	0.94	0.94	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97

Table 10 Correlations for the *rWIG*20 time series in the modified space

p	1	2	3	4	5	6	7	8	9	10
h_2	0.01	0.02	0.01	0.01	0.01	0.01	0.02	0.02	0.01	0.03
h_3	0.01	0.01	0.02	0.01	0.01	0.01	0.02	0.01	0.01	0.01
h_4	0.01	0.02	0.01	0.02	0.01	0.02	0.01	0.01	0.01	0.01

8 Summary

The article describes two new algorithms of the time series prediction. Both of them belong to the group of non-parametric methods and base on new definitions of neighborhood and locality. The prediction problem is brought to the estimation of the regression function in the modified data space.

The first algorithm (*HASKE*) defines the notion of *h*-neighborhood and helps to avoid the effect of its emptiness. Two required parameters of this algorithm are calculated adaptively with the usage of train and tune sets. The results of *HASKE* were compared with the results of the kernel regressor and the decomposition method. This algorithm is designed for time series that have well-defined time dependency (the value of the period is easy to interpret) like it is in the case of the *G* series (amount of passenger has the 12-month period). On the basis of this value the transformation to the new space is performed. Moreover, the data in the modified space should have significant correlation. If this condition is not fulfilled, the result of *HASKE* may not be satisfactory. It can be observed in the case of the results for *E* series prediction.

Second model (the *HKSVR*) is the local hybrid connection of the *SVR* and *HASKE*. It was tested on the real financial data. Generally, this algorithm is correct for time series that do not have the dominating period. That is the reason, why the other methods are used to find the value that defines the transformation of time series to the new space, for example the Fourier transformation. The algorithm improves the prediction for the short time horizons, excluding the next value prediction. Applicability of this model depends on the data correlation in the modified space. The normalized criterion $Q \in (0, 1)$ describing the improvement of prediction was used as a comparison tool. Values higher than 0.5 indicate that the usage of the *HKSVR* was appropriate.

References

1. Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Proceedings of the 5th annual workshop on computational learning theory, pp 144–152
2. Box GEP, Jenkins GM (1983) Time series analysis. PWN, Warsaw
3. Cleveland WS, Devlin SJ (1988) Locally weighted regression: an approach to regression analysis by local fitting. J Am Stat Assoc 403:596–610
4. de Boor C (2001) A practical guide to splines. Springer
5. Drucker H, Burges CJC, Kaufman L, Smola AJ, Vapnik VN (1997) Support vector regression machines. Adv Neural Inf Process Syst IX:155–161
6. Epanechnikov VA (1969) Nonparametric estimation of a multivariate probability density. Theory Probab Appl 14:153–158
7. Fan J, Gijbels I (1992) Variable bandwidth and local linear regression smoothers. Ann Stat 20:2008–2036
8. Fernández R (1999) Predicting time series with a local support vector regression machine. In: Proceedings of the ECCAI advanced course on artificial intelligence (ACAI '99)
9. Friedman JH (1991) Multivariate adaptive regression splines. Ann Stat 19:1–141
10. Gasser T, Kneip A, Kohler K (1991) A flexible and fast method for automatic smoothing. J Am Stat Assoc 415:643–652
11. Gasser T, Muller HG (1984) Estimating regression function and their derivatives by the kernel method. Scand J Stat 11:171–185
12. Hastie TJ, Tibshirani RJ (1990) Generalized additive models. Chapman & Hall/CRC
13. Hastie TJ, Tibshirani RJ, Friedman J (2001) The elements of statistical learning: data mining, inference, and prediction. Springer
14. Huang K, Yang H, King I, Lyu MR (2006) Local support vector regression for financial time series prediction. In: IJCNN'06 pp 1622–1627
15. Makridakis S, Wheelwright SC, McGee VE (1983) Forecasting: methods and applications, 2nd edn. Wiley, New York
16. Michalak M (2009) Time series prediction using new adaptive kernel estimators. Adv Intell Soft Comput 57:229–236
17. Nadaraya EA (1964) On estimating regression. Theory Probab Appl 9:141–142
18. Scholkopf B, Smola A (2002) Learning with Kernels. MIT Press
19. Scott DW (1992) Multivariate density estimation. Theory Pract Vis. Wiley & Sons
20. Silverman BW (1986) Density estimation for statistics and data analysis. Chapman & Hall
21. Smola AJ, and Scholkopf B (2004) A tutorial on support vector regression. Stat Comput 14:199–222
22. Taylor JS, Cristianini N (2004) Kernel methods for pattern analysis. Cambridge University Press
23. Terrell GR (1990) The maximal smoothing principle in density estimation. J Am Stat Ass 410:470–477
24. Terrell GR, Scott DW (1992) Variable kernel density estimation. Ann Stat 20:1236–1265
25. Turlach BA (1993) Bandwidth selection in kernel density estimation: a review. C.O.R.E. and Institut de Statistique, Universite Catholique de Louvain
26. Vapnik VN (1988) Statistical learning theory. Wiley
27. Wand MP, Jones MC (1995) Kernel smoothing. Chapman & Hall
28. Watson GS (1964) Smooth regression analysis. Sankhya—The Indian J of Stat 26:359–372
29. WIG 20 index closing values: http://stooq.pl/q/d/?s=wig20
30. Zelias P, Pawelek B, Wanat S (2004) Economic forecasting. Theory, examples, exercises. PWN, Warsaw